JURNAL DASI ISSN: 1411-3201
Vol. 11 No. 4 DESEMBER 2010

CONTOH IMPLEMENTASI DESIGN PATTERN

Armadyah Amborowati
STMIK AMIKOM Yogyakarta

Abstraksi

Design pattern merupakan komponen yang diperlukan dalam proses reusable-code pada pemrograman berorientasi objek. Berbagai jenis design pattern yang sering digunakan antara lain Singleton, Adapter Pattern, Façade Pattern, Bridge Pattern, Strategy Pattern, Observer Pattern, dan Tamplete Pattern. Dalam penulisan ini menjelaskan mengenai tujuan dari masing-masing design pattern dan contoh implementasinya.

Keywords: *Design Pattern*

Pendahuluan

Design Pattern merupakan komponen yang diperlukan dalam proses *reusable-code* pada pemrograman berorientasi objek. Berbagai jenis *design pattern* yang sering digunakan antara lain *Singleton, Adapter Pattern, Façade Pattern, Bridge Pattern, Strategy Pattern, Observer Pattern*, dan *Tamplete*

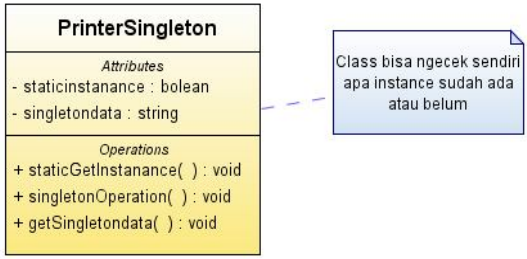
Pembahasan

Singleton

Tujuannya adalah membuat suatu *class* dengan method membuat instance baru (untuk memastikan apakah hanya ada sebuah instance suatu *class* diciptakan).

Contoh:

Meskipun ada banyak printer dan dokumen yang akan dicetak, tetapi hanya ada 1 printerSingleton. Class tersebut bertanggung jawab untuk mentracking instanance-nya sendiri.



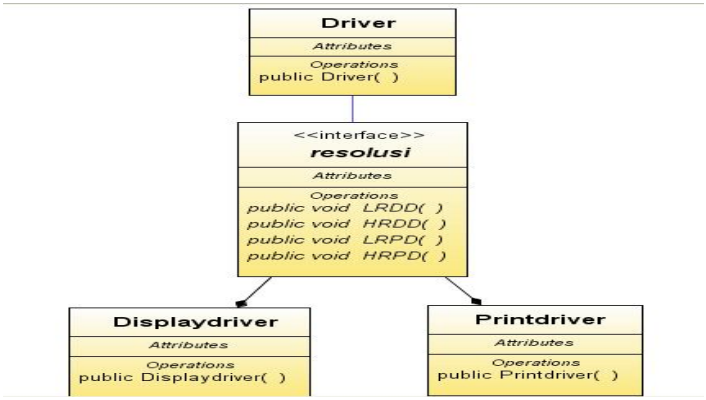
Gambar 1. Contoh Design Pattern Singleton

Adapter Pattern

Digunakan untuk membuat struktur data berorientasi obyek yang serbaguna. Membuat Interface/ class baru untuk menghubungkan 2 method yang berbeda. Caranya:

1. Definisikan sebuah kelas dasar.
2. Buat kelas struktur-data yang menyimpan kelas dasar tersebut.
3. Nanti si kelas-struktur data bisa menyimpan semua jenis kelas yang ditingkatkan (extend) dari kelas dasar, karena adanya sifat polymorphism.

Contoh:



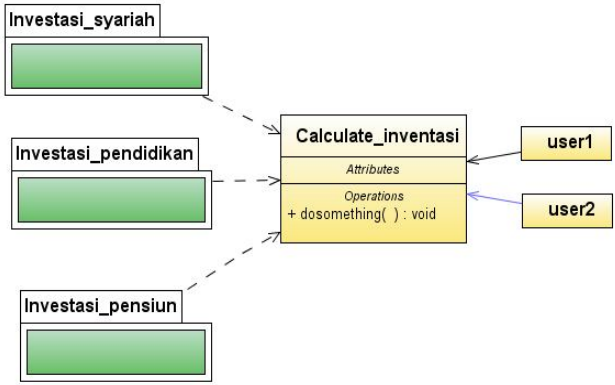
Gambar 2. Contoh Design Pattern Adapter Pattern

Façade Pattern

Digunakan untuk menyediakan interface agar subsistem mudah digunakan.

Contohnya:

Karena jenis investasi bermacam-macam maka untuk memudahkan pemakaian dibuat *façade object* (Calculate_invenstasi) yang menyediakan single interface sebagai bentuk simplikasi untuk mengakses fasilitas atau service yang disediakan oleh sebuah subsistem.



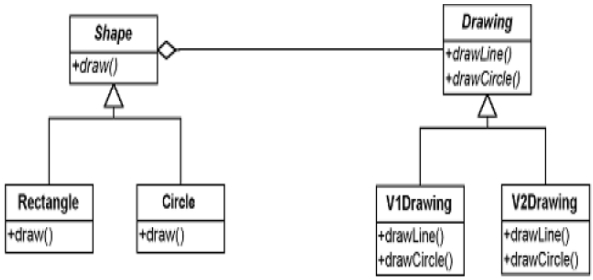
Gambar 3. Contoh Design Pattern Facade Pattern

Bridge Pattern

Digunakan untuk mencapsulate 2 aplikasi yang berbeda dengan memakai *Abstrak Class*.

Contoh:

Untuk menggambar kotak atau lingkaran maka dibutuhkan interface yang digunakan untuk memanggil algoritma 1 (V1Drawing) dan algoritma 2 (V2 Drawing).



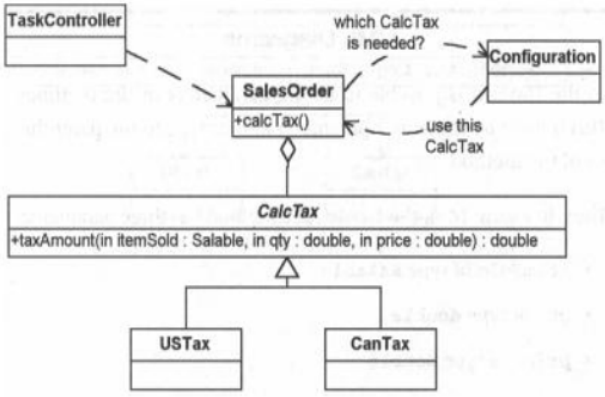
Gambar 4. Contoh Design Pattern Bridge Pattern (Sumber: Ridi, MTI,2008)

Strategy Pattern

Tujuannya adalah mendefiniskan suatu keluarga algoritma yang dipisahkan dari object yang asli guna meningkatkan fleksibilitas dan *reusability*. Penerapan *strategy pattern* ini memungkinkan client dapat menggunakan algoritma tersebut secara bergantian dengan bebas.

Contoh:

Pada salesorder perbedaan perhitungan pajak antara US dan Canada. Melalui Calctax client bisa secara bebas memilih US atau Canada.



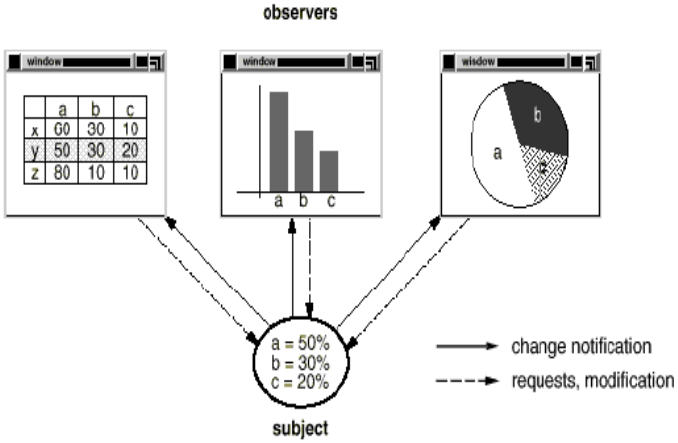
Gambar 5. Contoh Design Pattern Strategy Pattern (Sumber: Ridi, MTI,2008)

Observer Pattern

Tujuannya adalah mendefiniskan hubungan one-to-many antar object sehingga ketika sebuah object berubah state-nya, object-object lain yang bergantung juga ikut berubah.

Contoh:

Contoh pada aplikasi excel pada object table dan grafik. Jika ada perubahan pada table maka secara otomatis grafiknya juga ikut berubah.



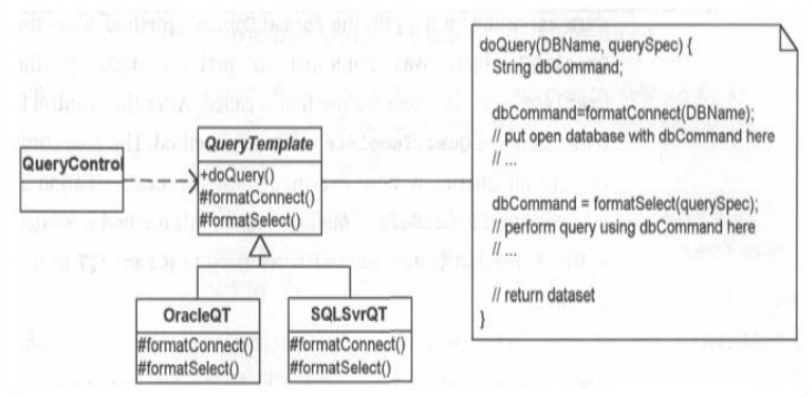
Gambar 6. Contoh Design Pattern Observer Pattern (Sumber: Gamma, 1995, h.249).

Tamplete Pattern

Tujuannya adalah membuat kerangka suatu algoritma.

Contoh:

Untuk melakukan query pada suatu DBMS seperti Oracle dan SQL server format koneksinya dan proses untuk Insert, update, delete, dan select sama maka dibuat suatu Tamplete untuk untuk melakukan proses tersebut.



Gambar 7. Contoh Design Pattern Tamplete Pattern (Sumber: Ridi, MTI,2008)

Penutup

Dalam penggunakan *design pattern* untuk *reusable-code* perlu diperhatikan fungsi dari masing-masing *design pattern*.

Daftar Pustaka

Ferdiana, Ridi, Materi object-oriented programming, MTI, 2008.

Gamma, Erich, *Design Pattern: Element of reusable object-oriented software*, Addison-wisley, 1995.